Evolving Evolvability in the Context of Environmental Change: A Gene Regulatory Network (GRN) Approach

Yifei Wang[1], Stephen G. Matthews[2] and Joanna J. Bryson[1]

[1]Intelligent Systems Group, Department of Computer Science, University of Bath, Bath, BA1 7AY, U.K.
[2]Intelligent Systems Laboratory, Department of Engineering Mathematics, University of Bristol, Bristol, BS8 1UB, U.K.
yifei.wang@ieee.org, stephen.matthews@bristol.ac.uk, j.j.bryson@bath.ac.uk

Abstract

Evolvability is the capacity of a genotype to rapidly adjust to certain types of environmental challenges or opportunities. This capacity, documented in nature, reflects foresight enabled by the capacity of evolution to capture and represent regularities not only in extant environments, but in the ways in which the environments tend to change. Here we posit that evolvability substantially benefits from the hierarchical representations afforded by Gene Regulatory Networks (GRNs). We present an extension of standard Genetic Algorithms (GAs) and demonstrate its capacity to learn a genotype phylogeny able to express rapid phenotypic shifts in the context of an oscillating environment.

Introduction

GAs are methods well-suited for search and optimisation in non-linear and high-dimensional problems. Convergence to near-optimal solutions is often perceived as the goal for GAs. However, overspecialisation can produce fragile solutions. In nature, selection also sustains *evolvability*, the capacity to change. The simplest form of evolvability is simply variation—the rate of evolution is determined by the amount of variation in a population (Fisher, 1930; Price, 1972; Reisinger and Miikkulainen, 2006; Hu and Banzhaf, 2010). More specialised evolvability can be achieved via hierarchical representations, for example single-gene control of beak length in Darwin's finches (Campàs et al., 2010). Evolvability allows rapid adaptation to regular changes in the environment, whilst also preventing premature convergence to local optima.

Here we present a system for discovering highly-evolvable genomes by exploiting GRNs (van Dijk et al., 2012; Payne et al., 2013). GRNs are already known to produce robustness via evolvability (Aldana et al., 2007a; Crombach and Hogeweg, 2008). The many-to-one mapping mechanism of genotype to phenotype implicit in GRNs enables genes to buffer against and even exploit likely variations in the genome. In addition, such a dual learning system—coupled plasticity—is known to accelerate evolution in the right contexts (Hinton and Nowlan, 1987; Kashtan et al., 2007; Borenstein and Krakauer, 2008). Hinton and Nowlan (1987) focus on the interaction between evolution and learning, showing that coupled plasticity can solve a problem that is extremely difficult for an evolutionary process on its own. Our aim and approach here are similar, but our mechanism and outcome are novel. We explore how the robustness of GRNs can improve the evolvability of GAs by exploiting GRNs for learning structure required for quick adaptations to environmental change. This paper contributes: 1) a biologically-motivated GRN model and associated understanding of its dynamics, and 2) a demonstration of learning a quick and robust response to a changing environment—in other words, improved evolvability.

Related Work

We introduce some fundamental biological aspects of GRN, and discuss several GAs that have similar goals and properties.

Gene Regulatory Networks

GRNs control the expression of genes for providing phenotypic traits in living organisms. They play a central role in cells and govern cell differentiation, metabolism, the cell cycle, and signal transduction (Karlebach and Shamir, 2008). A GRN contains a network of genes (regions of DNA sequences) with interaction mechanisms for controlling gene expression. Epigenetic factors control the regulation of gene expression in the network without changing the genes. Shifts in gene regulation provide plasticity where organisms can adapt to environmental change. The presence of genes in the network and interactions between genes creates this plasticity.

There are a number of prior studies exploring both theoretical and practical aspects of GRNs from a biological perspective. Aldana et al. (2007b) studied the robustness and evolvability of the attractor landscape of GRNs under the process of gene duplication followed by divergence. Balleza et al. (2008) showed the criticality coupled in GRNs can generate the great diversity of dynamically robust living forms. Crombach and Hogeweg (2008) demonstrated that

long-term evolution of complex GRNs in a changing environment can increase the efficiency of generating beneficial mutations.

Network Structures in Evolutionary Algorithms

Graph structures are common in Evolutionary Algorithms (EAs), such as undirected acyclic graphs (trees) in genetic programming (Koza, 1989) and cyclic graphs in evolutionary programming (Fogel, 1962). Genetic programming and grammatical evolution (Ryan et al., 1998) contain redundant genes in the chromosome, allowing some unexpressed genetic variation to be carried through generations. This redundancy can act like a memory system to enhance evolvability in dynamic environments. For example, Goldberg and Smith (1987) explored a diploid GA (with dominance operators) where a gene contains two alleles. A diploid GA was shown to adapt quicker to environmental change than a haploid (one allele per gene) GA. As well as redundancy, indirect encodings are beneficial for improving evolvability and they more closely represent the robustness and complexity found in nature (Reisinger and Miikkulainen, 2006; Reisinger et al., 2005).

Until recently, EAs had not incorporated the epigentics of GRN (Hu and Banzhaf, 2010). Lopes and Costa (2013a,b) use a model of GRN, which they refer to as an artificial regulatory network (ARN), to exploit epigenetics in genetic programming and grammatical evolution for the inverted pendulum problem, drawing artificial art, and the artificial ants problem (Sante Fe Trail problem). GRN in an EA has also been applied to financial trading (Nicolau et al., 2012). None of these GRN-based approaches to machine learning with EAs specifically address evolvability and the use of GRNs for handling environmental change. However, several authors have suggested there may be benefits to the approach (Hu and Banzhaf, 2010; Lopes and Costa, 2013a).

Model: GRN with Sexual Reproduction

The GRN with Sexual Reproduction model extends the standard GA with a GRN representation and associated reproduction and mutation operations which are different from the standard GA (see more details below). A simple model GRN provides a mechanism for improving evolvability: The interaction network is a multi-layer learning system. During the evolution process we used sexual rather than asexual reproduction in GRN because in tests it achieved better performance [1].

GRN with Sexual Reproduction models the evolutionary process in the following steps, which are detailed in pseudocode in Algorithm 1. A large number of gene regulatory networks with a certain level of connectivity c are generated randomly; of these stable networks are selected. Then, all the stable networks are re-selected based on their fitness. A certain number of networks are mutated according to the given mutation rate, P_m. Finally, all networks in the population undergo free combination[2] (Azevedo et al., 2006) during sexual reproduction.

Artificial Gene Regulatory Network

The GRN model from Wagner (1996); Siegal and Bergman (2002); Azevedo et al. (2006) is well established and provides the basic model of gene interactions required to demonstrate its capacity to learn an evolvable genotype.

For each individual in a finite population M, an $N_{gene} \times N_{gene}$ matrix W is an artificial gene network that contains the regulatory interactions among N_{gene} genes. An example is given in Figure 1A. Each element $w_{i,j}$ $(i, j = 1, 2, \ldots, N_{gene})$ represents the regulatory effect on the expression of gene i of the product of gene j (see Figure 1B). The connectivity parameter c determines the proportion of non-zero elements in the network W. Through gene interactions, the regulatory effect acts on each gene expression pattern (in network W) are denoted by a state vector $\mathbf{S}(t) = (s_1(t), s_2(t), \ldots, s_{N_{gene}}(t))$ where $s_i(t)$ represents the expression pattern of gene i at time t. Each value of expression state $s_i(t)$ is within the interval $[-1, 1]$ that expresses complete repression (-1) and complete activation $(+1)$. For a given gene regulatory network W, the dynamics of $\mathbf{S}$ for each gene i is modelled by

$$s_i(t+1) = f\left(\sum_{j=1}^{N} w_{ij} s_j(t)\right), \tag{1}$$

where $f(x)$ is a sigmoidal function. In this paper, we defined $f(x) = 2/(1 + e^{-ax}) - 1$, where a is a free parameter determining the rate of change from complete repression to complete activation. When a is large enough, for example $a = 1,000$, $f(x)$ is degenerated similarly as a sign function where $f(x) = -1$ for $x < 0$, $f(x) = +1$ for $x > 0$ and $f(0) = 0$.

In all simulations, we defined the network developmental stability as the progression from an initial expression state to an equilibrium expression state (reaching a fixed pattern) by iterating Equation (1) within a fixed number of times, $devT$. A simple example of iterating Equation (1) is given in Figure 1C. If a given network W can achieve developmental stability, it is termed as viable or stable network, otherwise it is labelled unviable or unstable. We determined that an equilibrium expression state can be reached when the following equation is met

$$\frac{1}{\tau} \sum_{\theta=t-\tau}^{t} D\left(\mathbf{S}(\theta), \overline{\mathbf{S}}(t)\right) \leq 10^{-4}, \tag{2}$$

[1] We did the same simulation with asexual GRN model, and found it was evolved very slow (the fitness was only improved a little during each shifting cycle) compared with sexual GRN model.

[2] For more details of implementation of free combination see the cited paper.

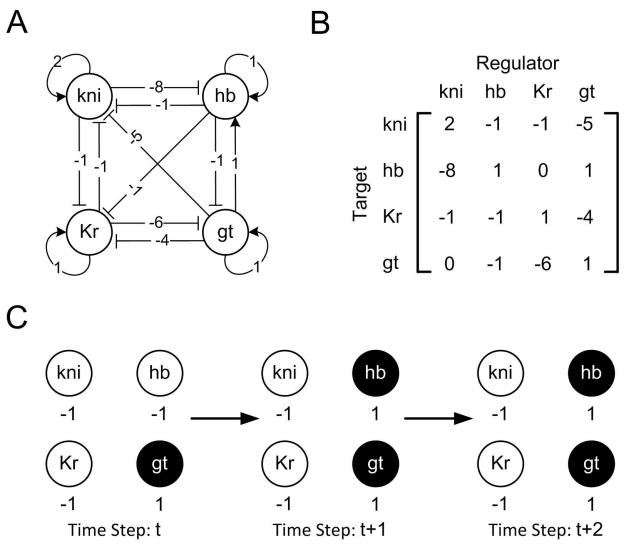

Figure 1: Example of gene regulatory network model in the gap gene system of *Drosophila melanogaster*, reproduced from Azevedo et al. (2006). (A), Network representation of the regulatory iteration between four gap genes (giant (gt), hunchback (hb), knirps (kni), Krüppel (Kr)) (Crawley, 2002). Gene activations and repressions are denoted by arrows and bars, respectively. Numbers on each directional edge indicate the strength (weight) of interaction between the two linked genes (Jaeger et al., 2004). (B), Interaction matrix (W) represents the network in A. Each element in row i and column j, i.e., w_{ij}, represents the regulatory effect on the expression of gene i of the product of gene j. (C), Graphical representation of the gene expression states of each gap gene over three successive time steps. We use -1 (unfilled circle) to denote the complete repression state if $s_i(t) \leq 0$, and 1 (filled circle) to represent complete activation state if $s_i(t) > 0$. Therefore, the state vectors $[kni, hb, Kr, gt]$ of four gap genes at time t, $t+1$, and $t+2$ are $\mathbf{S}(t) = (-1,-1,-1,1)$, $\mathbf{S}(t+1) = (-1,1,-1,1)$, and $\mathbf{S}(t+2) = (-1,1,-1,1)$, respectively. Successive iterations beyond the $t+2$ step do not change the gene expression pattern, which is the hallmark of a stable equilibrium.

where $D(\mathbf{S}, \overline{\mathbf{S}}) = \sum_{i=1}^{N_{gene}} (s_i - s'_i) \Big/ 4N_{gene}$ measures the difference between the gene expression pattern $\mathbf{S}$ and $\overline{\mathbf{S}}$, and $\overline{\mathbf{S}}$ is the average of the gene expression levels over the time interval $[t-\tau, t-\tau+1, \ldots, t]$. Unless otherwise specified, we used $devT = 100$, $\tau = 10$, and $a = 2$ in all simulations.

Initialisation

Each individual in population M was generated with a gene regulatory network W associated with an expression state vector $\mathbf{S}(0)$. The network was generated by randomly filling W with $c \cdot N_{gene}^2$ non-zero elements $w_{i,j} \sim N(0,1)$. The associated initial expression state $\mathbf{S}(0)$ was randomly setting each $s_i(0)$ to -1 or 1.

Selection

Competitive selection occurs using roulette-wheel selection. Survivors are placed in a new population.

Mutation

Offspring are generated by picking an individual at random from the population and placing in the new population. Then, in the selected network, each non-zero entry in the W interaction matrix was replaced by $w'_{i,j} \sim N(0,1)$ with probability P_m. Note that mutations should be viewed as operating on the $c \cdot N_{gene}^2$ cis-regulatory elements, not the coding sequences of the N_{gene} genes themselves. In other words, the mutation operation cannot change the topology of the original network W. Only offspring that are capable of producing a stable gene expression pattern survive. This process is repeated until the same amount of developmentally stable networks are produced.

Sexual Reproduction

Offspring are generated by picking two individuals at random from the population. Then the chosen two networks undergo sexual reproduction, selecting rows of the W matrices from each parent with equal probability. This process is similar to free recombination between units formed by each gene and its cis-regulatory elements, but with no recombination within regulatory regions.

Phenotype and Fitness

The fitness of an individual is determined by its phenotype. Here this is the subset of its genotype that is expressed. While in Nature the GRN might determine any number of traits to be expressed, here the fitness function considers only a fixed number (N_{PhT}) of the most upregulated genes. Level of regulation is determined by the GRN as per Equation 1. For more details of implementation of above operations please refer to Algorithm 1.

Experiments

Approach

As previously reviewed, evolvability is facilitated by redundant encoding which maintains variation, allowing quick changes to be made between good solutions. Here we have not only dual chromosomal structure, but gene activation driven by the GRN. Neither redundancy nor evolvability more generally necessarily improve EAs—in some cases these can slow evolution (Reisinger and Miikkulainen, 2006). As with all evolution and learning more generally, a gradient is required.

We compared the GRN with Sexual Reproduction model with a standard GA (Holland, 1975) to assess its capability

Algorithm 1 GRN with Sexual Reproduction Evolution Process

procedure GRN WITH SEXUAL REPRODUCTION
  set: c, $devT$, τ, a ▷ Initialise GRN with Sexual Reproduction model parameters
  set: Pop_{size}, N_{gene}, N_{PhT}, P_m, G_{shift}, g and $gMax$ ▷ Initialise evolution parameters
  create: $\{GRN_{pop}(n); n = 1, 2, \ldots, Pop_{size}\}$ ▷ Randomly generate stable networks, using c, $devT$, τ, and a
    ▷ An individual network is associated with regulatory matrix $W(n)$ and state $\mathbf{S}(n)$
  for $g \leftarrow 1, gMax$ **do**
    calculate: $\{Fit_{pop}(n); n = 1, 2, \ldots, Pop_{size}\}$ ▷ Evaluate the fitness of each individual in the population
    while $n < Pop_{size}$ **do**
      select: $GRN_{pop}(p)$ $(p \in 1, 2, \ldots, Pop_{size})$ ▷ Select an individual based on its fitness
      save: $GRN_{pop}(n) \leftarrow GRN_{pop}(p)$ ▷ Save the selected individual into population
      $n \leftarrow n + 1$
    end while
    while $n < Pop_{size}$ **do**
      pick: $GRN_{pop}(p)$ $(p \in 1, 2, \ldots, Pop_{size})$ ▷ Randomly pick one individual
      if $rand(0, 1) < P_m$ **then** ▷ A non-zero item in $W(p)$ is mutated with probability P_m
        mutate: $w_{i,j} \leftarrow N(0, 1)$ ▷ The non-zero item in $W(p)$ is replaced by $N(0, 1)$
      end if
      if $GRN_{pop}(p)$ is stable **then** ▷ Only stable networks survive
        save: $GRN_{pop}(n) \leftarrow GRN_{pop}(p)$ ▷ Save the stable individual into population
        $n \leftarrow n + 1$
      end if
    end while
    while $n < Pop_{size}$ **do**
      pick: $GRN_{pop}(p), GRN_{pop}(q)(p, q \in 1, 2, \ldots, Pop_{size})$ ▷ Randomly pick two individuals
      recombine: $GRN'_{pop}(p), GRN'_{pop}(q)$ ▷ New individuals are generated by free combination
      if $GRN'_{pop}(p)$ and/or $GRN'_{pop}(q)$ are stable **then** ▷ Only stable networks survive
        save: $GRN_{pop}(n) \leftarrow GRN_{pop}(p)$ and/or $GRN_{pop}(q)$ ▷ Save the stable individual(s) into population
        $n \leftarrow n + 1$ or $n \leftarrow n + 2$ ▷ If both stable $+2$, otherwise $+1$
      end if
    end while
    if $\mathrm{mod}(g, G_{shift}) = 0$ **then** ▷ Test if g reach the new cycle of shifting generation
      switch fitness function ▷ The environment is changed in every G_{shift} generations
    end if
  end for
end procedure

and potential benefits for machine learning in environments that change. Here, we apply selective pressure for evolvability by changing the fitness function. We create a modified version of MaxOnes, called MaxABs. This assigns fitness according to the proportion of expressed genes that conform to the current environmental expectation, which switches between A and B after a set number of generations, which is defined by the parameter G_{shift} (see also Kashtan and Alon, 2005). For example, the first 20 generation's fitness is $\frac{number\ of\ As}{number\ of\ As+Bs}$ in the phenotype, the next 20 is $\frac{number\ of\ Bs}{number\ of\ As+Bs}$, the next 20 reverts to $\frac{number\ of\ As}{number\ of\ As+Bs}$, and so on. Note that for fitness, we only count As and Bs that are in the *expressed* genes.

Three experiments were conducted with mostly the same model parameters, but different chosen N_{gene}, N_{PhT} and time lags (G_{shift}) between environmental changes.

Parameters

Unless specified, the mutation rate is fixed at $P_m = 0.001$ for the GA and GRN with Sexual Reproduction model in all simulations. For the GA, the recombination probability P_r is fixed to be 0.7, and for the GRN with Sexual Reproduction model, free recombination is used. Network connectivity c was set to small (details below) for all runs of the GRN with Sexual Reproduction model. This is because theory shows that selection for robustness will favor more sparsely connected and minimally complex networks (Leclerc, 2008). Roulette wheel selection is used in both approaches. In all experiments, the population size was $Pop_{size} = 50$. Unless specified, 500 independent runs were conducted.

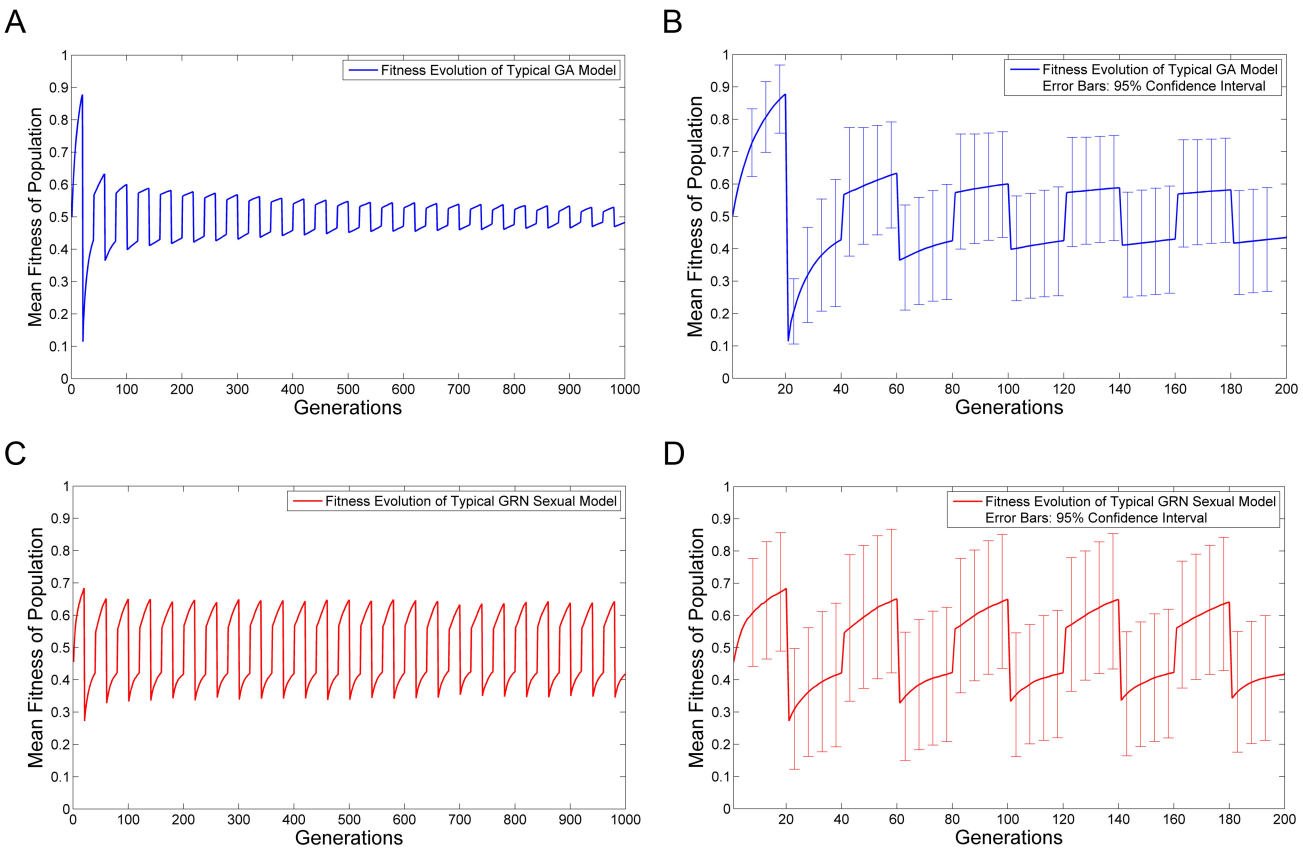

Figure 2: Evolution of both algorithms with small time gaps ($G_{shift} = 20$) between environmental change for typical GA model (A) and typical GRN with Sexual Reproduction model (C). The first 200 generations of GA and GRN with Sexual Reproduction model are shown in (B) and (D), respectively. Error bars indicate 95% confidence interval based on 500 independent runs.

Small Time Gaps Between Environmental Change

The fitness function was configured to switch between As and Bs every 20 generations to assess evolvability in frequently changing environments. The algorithms were terminated at $1,000$ generations, the number of genes was $N_{gene} = 20$, of which $N_{PhT} = 10$ occur in the phenotype, and network connectivity was $c = 0.25$. The mean fitness values in each population for the GA and GRN with Sexual Reproduction model are shown in Figure 2. Two observations are made.

The first observation is that the mean fitness increases to about 0.9 (for MaxAs) in the GA model, whereas it increases to about 0.7 in the GRN with Sexual Reproduction model. But for both models, the fitness does not reach the same magnitude of mean fitness in the subsequent MaxAs targets.

The second observation is that after the first two changes in fitness function (one for MaxAs and one for MaxBs) the GRN with Sexual Reproduction model has higher mean fitness than the GA. This is especially clear when the GA evolves in more cycles. The GA's mean fitness is less and this suggests the GA is less evolvable than the GRN when switching between MaxAs and MaxBs.

Note that the goal of optimisation is to maximise the fitness value which is switching between MaxAs and MaxBs. Shifting from one peak with relatively high fitness to an even higher fitness peak is usually much more difficult than shifting from a low fitness stage to a higher fitness peak. In Figure 2, it seems that the GRN with Sexual Reproduction model has higher fitness for MaxAs and worse fitness for MaxBs. The reason is because their starting points are different. In fact, the relative increase of fitness is the same between MaxAs and MaxBs. In our simulation, we did observe that if we would give MaxBs enough time, then fitness of MaxBs can be increased to be the same level as the fitness of MaxAs.

Large Time Gaps Between Environmental Change

Based on the observation from the results produced from a small time gap, the fitness function was increased to switch between MaxAs and MaxBs every 200 generations

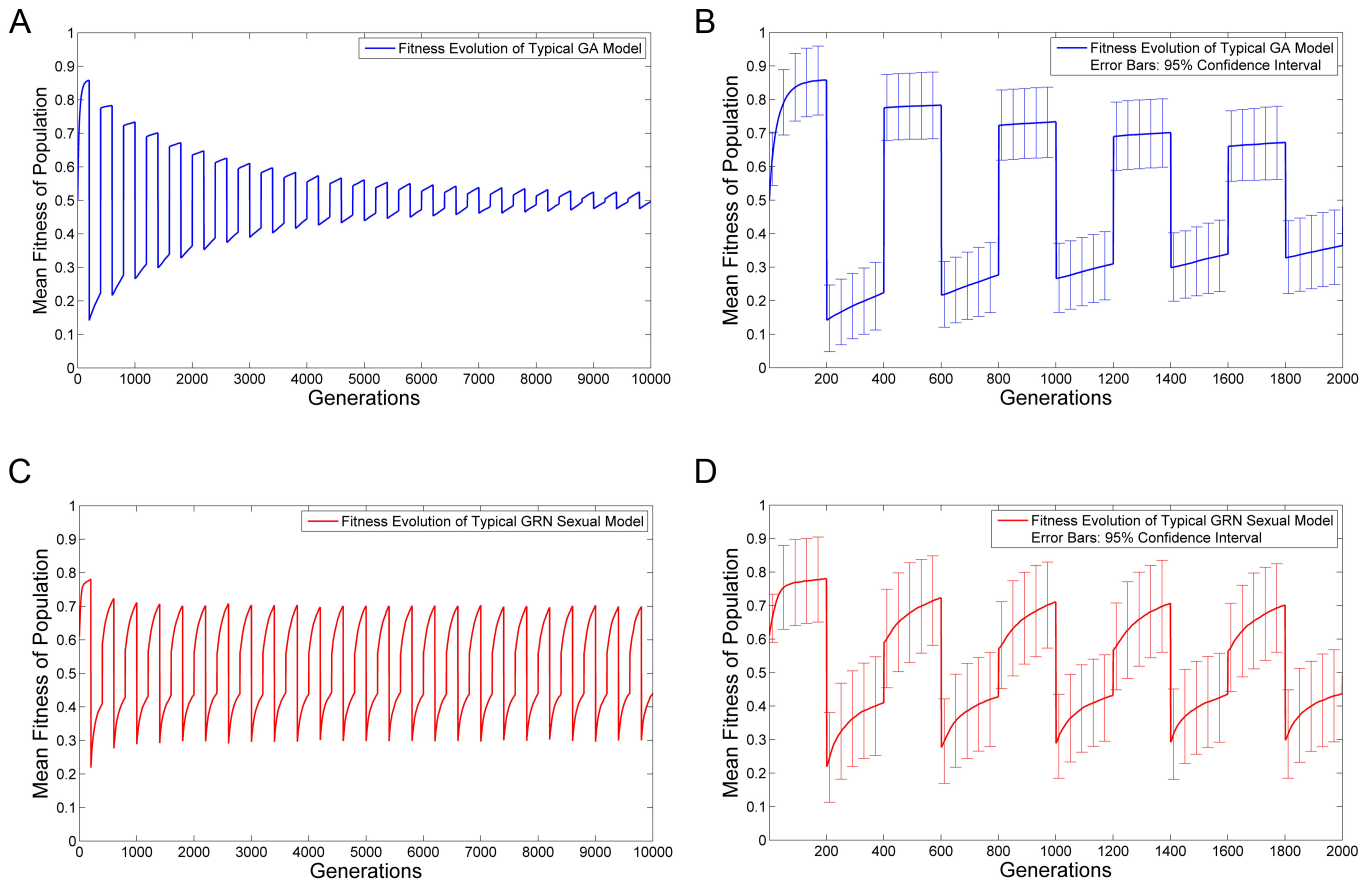

Figure 3: Evolution of both algorithms with large time gaps ($G_{shift} = 200$) between environmental change for typical GA model (A) and typical GRN with Sexual Reproduction model (C). The first $2,000$ generations of GA and GRN with Sexual Reproduction model are shown in (B) and (D), respectively. Error bars indicate 95% confidence interval based on 500 independent runs.

and the algorithms ran for longer by terminating at $10,000$ generations. We set the phenotype to be the entire genotype, $N_{gene} = N_{PhT} = 30$, and network connectivity $c = 0.05$. The purpose is to assess evolvability in less frequently changing environments, but also to check the impact of redundant encoding (see experiment 3 below).

The mean fitness values in each population for the GA and GRN with Sexual Reproduction model are shown in Figure 3. Four observations are made.

The first observation is that the mean fitness of both algorithms is largest in the first time gap, which is the same for both short and long time gaps.

The second observation is that the GA has a higher mean fitness in the first time gap than the GRN with Sexual Reproduction model. This suggests the GA performs better for MaxABs *without* environmental change and the GRN with Sexual Reproduction model performs worse (perhaps learns more slowly) in a static environment.

The third observation is that the GA's mean fitness diminishes gradually as the fitness function is switched (Figure 3A). However, the GRN with Sexual Reproduction model achieves the same mean fitness every time the fitness function is switched (Figure 3C). This further supports that the GA is less evolvable than the GRN with Sexual Reproduction model, because the GA is less able to adapt to a regular environmental change.

The fourth observation is that the GRN with Sexual Reproduction model has steeper gradients of mean fitness in each time gap (Figure 3D). This suggests the GRN with Sexual Reproduction model is more evolvable than the GA.

Advantages of Redundant Mapping in GRN

In the first experiment, there was a redundant mapping in the GRN, but we purposely set no redundant mapping in GRN in the second experiment. Both experiments show that the GRN with Sexual Reproduction model is capable of evolving in a changing environment. However, we didn't clearly observe any advantage of redundant mapping embedded in

the GRN in short time lags. Therefore, we conducted an additional experiment to further test whether there would be a benefit of many-to-one mapping of genotype to phenotype in GRN in longer time lags.

In this experiment, the fitness function switched between MaxAs and MaxBs every 200 generations and the algorithm terminated at $5,000$ generations. We set the number of genes in the genotype to $N_{gene} = 20$, of which $N_{PhT} = 10$ are expressed in the phenotype for selection, and network connectivity $c = 0.25$. The result is shown in Figure 4 based on 250 independent runs. Compared to the previous experiments, two observations are made.

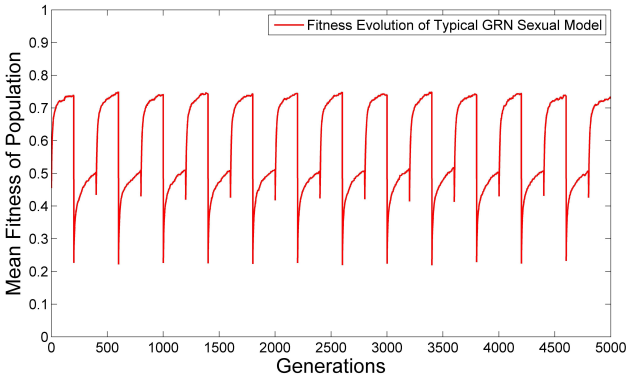

Figure 4: Evolution of typical GRN with Sexual Reproduction model with redundant mapping and large time gaps ($G_{shift} = 200$) between environmental change.

The first observation is that the GRN with Sexual Reproduction model fully recovers from the change in environment: the mean fitness in subsequent cycles can be restored to almost the same magnitude as the first cycle.

The second observation is that the GRN with Sexual Reproduction model can adapt to new environment much faster compared to Figure 2. This is particular clear when the environment switched from MaxBs to MaxAs. We further tracked how many generations would the GRN with Sexual Reproduction model be able to reach the set value of fitness[3] in the first and the last cycle of MaxAs and MaxBs during the evolution. We found the GRN with Sexual Reproduction model took 150 and 103 generations on average to reach the set fitness values for MaxAs and MaxBs, respectively, in the first cycle, where as it only took 119 and 48 generations on average to reach the set fitness values in the last cycle. This is clear evidence of evolvability.

[3]We set the test fitness values to be 0.6 for MaxAs, and 0.4 for MaxBs. Then we recorded the generation that the set values can be reached in 5 successive runs in a typical cycle.

Summary & Discussion

In this paper, we show that the typical GRN with Sexual Reproduction model is capable of evolving evolvability, and thus evolving faster than the typical GA model in a changing environment. We designed two situations to mimic the changing environment, forcing the population in GA and GRN with Sexual Reproduction model to evolve in shorter and longer time lags. We found that the evolvability of the GA model decreases as it tends towards a mean value for both environments. In contrast, the GRN with Sexual Reproduction model is able to continuously evolve in both cases, retaining peak performance. In addition, although the GA model converges faster than the GRN with Sexual Reproduction model, this convergence results in a lack of diversity and evolvability. This result shows the potential advantage of using hierarchical structures such as the GRN with Sexual Reproduction model to solve problems of dynamic environments that nevertheless vary in structured ways—a task that less-structured GAs are not equipped to address.

References

Aldana, M., Balleza, E., Kauffman, S., and Resendiz, O. (2007a). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245(3):433–448.

Aldana, M., Balleza, E., Kauffman, S., and Resendiz, O. (2007b). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245(3):433–448.

Azevedo, R. B. R., Lohaus, R., Srinivasan, S., Dang, K. K., and Burch, C. L. (2006). Sexual reproduction selects for robustness and negative epistasis in artificial gene networks. *Nature*, 440(7080):87–90.

Balleza, E., Alvarez-Buylla, E. R., Chaos, A., Kauffman, S., Shmulevich, I., and Aldana, M. (2008). Critical dynamics in genetic regulatory networks: Examples from four kingdoms. *PLoS ONE*, 3(6):e2456.

Borenstein, E. and Krakauer, D. C. (2008). An end to endless forms: Epistasis, phenotype distribution bias, and non-uniform evolution. *PLoS Computational Biology*, 4(10):e1000202.

Campàs, O., Mallarino, R., Herrel, A., Abzhanov, A., and Brenner, M. P. (2010). Scaling and shear transformations capture beak shape variation in Darwin's finches. *PNAS*, 107(8):3356–3360.

Crawley, M. (2002). *Statistical Computing: An Introduction to Data Analysis Using S-Plus*. Wiley.

Crombach, A. and Hogeweg, P. (2008). Evolution of evolvability in gene regulatory networks. *PLoS Computational Biology*, 4(7):e1000112.

Fisher, R. A. (1930). *The Genetical Theory of Natural Selection.* Oxford-University-Press.

Fogel, L. J. (1962). Autonomous automata. *Industrial Research*, 4(2):14–19.

Goldberg, D. E. and Smith, R. E. (1987). Nonstationary function optimization using genetic algorithm with dominance and diploidy. In *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, pages 59–68.

Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1(3):495–502.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press.

Hu, T. and Banzhaf, W. (2010). Evolvability and speed of evolutionary algorithms in light of recent developments in biology. *Journal of Artificial Evolution and Applications*, 2010:1–28.

Jaeger, J., Blagov, M., Kosman, D., Kozlov, K. N., Manu, Myasnikova, E., Surkova, S., Vanario-Alonso, C. E., Samsonova, M., Sharp, D. H., and Reinitz, J. (2004). Dynamical analysis of regulatory interactions in the gap gene system of drosophila melanogaster. *Genetics*, 167(4):1721–1737.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol*, 9(10):770–780.

Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *PNAS*, 102(39):13773–13778.

Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104(34):13711–13716.

Koza, J. R. (1989). Hierarchical genetic algorithms operating on populations of computer programs. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'89, pages 768–774, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Leclerc, R. D. (2008). Survival of the sparsest: Robust gene networks are parsimonious. *Molecular Systems Biology*, 4(1).

Lopes, R. L. and Costa, E. (2013a). Gearnet: Grammatical evolution with artificial regulatory networks. In *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, pages 973–980.

Lopes, R. L. and Costa, E. (2013b). Genetic programming with genetic regulatory networks: Genetic programming. In *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, pages 965–972.

Nicolau, M., Oeill, M., and Brabazon, A. (2012). Applying genetic regulatory networks to index trading. In Coello, C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., and Pavone, M., editors, *Parallel Problem Solving from Nature - PPSN XII*, volume 7492 of *Lecture Notes in Computer Science*, pages 428–437.

Payne, J. L., Moore, J. H., and Wagner, A. (2013). Robustness, evolvability, and the logic of genetic regulation. *Artificial Life*, 20(1):111–126.

Price, G. R. (1972). Fisher's 'fundamental theorem' made clear. *Annals of Human Genetics*, 36(2):129–140.

Reisinger, J. and Miikkulainen, R. (2006). Selecting for evolvable representations. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 1297–1304.

Reisinger, J., Stanley, K. O., and Miikkulainen, R. (2005). Towards an empirical measure of evolvability. In *Proceedings of the 2005 Workshops on Genetic and Evolutionary Computation*, GECCO '05, pages 257–264.

Ryan, C., Collins, J., and O'Neill, M. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In Banzhaf, W., Poli, R., Schoenauer, M., and Fogarty, T., editors, *Genetic Programming*, volume 1391 of *Lecture Notes in Computer Science*, pages 83–96. Springer Berlin Heidelberg.

Siegal, M. L. and Bergman, A. (2002). Waddington's canalization revisited: Developmental stability and evolution. *PNAS*, 99(16):10528–10532.

van Dijk, A. D. J., van Mourik, S., and van Ham, R. C. H. J. (2012). Mutational robustness of gene regulatory networks. *PLoS ONE*, 7(1):e30591.

Wagner, A. (1996). Does evolutionary plasticity evolve? *Evolution*, 50(3):1008–1023.